

Aula Prática: Algoritmos de Ordenação

Objetivo: *Exercitar o desenvolvimento de algoritmos de ordenação, capacitando o estudante a entender e implementar tanto os algoritmos simples, quanto os mais eficientes.*

Exercício:

Implemente um programa que carregue um vetor com as informações disponibilizadas em um arquivo texto. Cada linha do arquivo representa um estudante e é formada por 5 dados, devidamente separados por espaço: matrícula (inteiro), nome (string), e 3 notas (float). Recomenda-se o uso da função `fscanf` para facilitar a leitura desses campos.

Após o carregamento, o programa deve ordenar o vetor resultante, usando o campo de matrícula como chave e o algoritmo selecionado pelo usuário (1- *Bubble Sort*, 2- *Select Sort*, 3- *Insert Sort*, 4- *Quick Sort* ou 5- *Merge Sort*). Como saída, será exibida a lista de alunos ordenada; as quantidades de comparações e trocas realizadas pelo algoritmo durante a ordenação (obs: esses dados devem ser contabilizados e retornados por cada algoritmo de ordenação); e o tempo de execução da ordenação (indicando a unidade de medida utilizada – minutos, segundos ou milissegundos).

OBS: *para contabilizar o tempo de execução, recomenda-se o uso da biblioteca `time.h`, a qual possui um tipo de dado específico e funções capazes de capturar os tempos de início e fim da execução, bem como calcular a diferença entre eles.*

Por fim, utilize o programa implementado para construir uma tabela no excel, com o número de comparações e trocas, bem como o tempo de execução que cada algoritmo implementado gastou para ordenar o arquivo texto em anexo (**Lista_Alunos.txt**), o qual contém 2287 registros de estudantes.